

# APLIKASI GENERATOR SCRIPT BASIS DATA BERORIENTASI OBYEK ORACLE 10g

Titus Adi Prasetyo<sup>(1)</sup>

## Abstrak:

Dengan berkembangnya pemodelan basis data yang di dasarkan pada konsep pemodelan berbasis obyek, maka diperlukan aplikasi yang mampu digunakan untuk memodelkan basis data berorientasi obyek yang selama ini belum pernah dibangun. Aplikasi yang dibangun di beri nama Rum dan mampu digunakan untuk memodelkan basis data berorientasi obyek dan melakukan generate pemodelan tersebut ke dalam SQL di Oracle 10g. Analisis dilakukan pada hasil kuesioner yang disebarkan kepada beberapa tester yang mencoba menggunakan Rum. Kesimpulan yang diperoleh antara lain, SQL hasil generate Rum mampu di eksekusi dengan benar di basis data Oracle 10g. Dan yang kedua berdasarkan respon pengguna yang melakukan testing terhadap Rum, bahwa Rum memiliki kelemahan di sisi interface. Dan yang terakhir adalah pengguna yang memiliki latar belakang pengetahuan pemodelan berbasis obyek dan basis data Oracle 10g yang baik akan sangat membantu menggunakan Rum.

**Kata Kunci :** *basis data, oracle, object relational database*

## 1. Pendahuluan

Basis data yang digunakan dalam suatu aplikasi bisnis pada umumnya dirancang dengan menggunakan pendekatan *Relational Database Management System (RDBMS)*, tetapi di lain pihak pada sisi *programming* aplikasi dibangun dengan menggunakan pendekatan *Object-Oriented Programming* (Pemrograman Berorientasi Obyek). Ini menyebabkan terjadinya apa yang disebut *Impedance Mismatch* (ketidaksesuaian) yaitu data yang disimpan dalam basis data tidak konsisten dengan model pemrograman yang dipakai yaitu pemrograman berorientasi obyek karena pada pendekatan pemrograman berorientasi obyek, data diperlakukan sebagai kumpulan obyek yang dipaksa dimasukkan ke dalam basis data yang tidak menerapkan pendekatan *object-oriented*.

Salah satu model basis data yang dapat digunakan untuk menjembatani model berbasis obyek adalah *object relational database*. Dengan model ini, perancangan suatu basis data dapat dilakukan berdasar pendekatan konsep berorientasi pada obyek.

Untuk mempermudah dalam merancang dan menerapkan konsep berbasis obyek pada *object relational database*, perlu disediakan sebuah alat bantu perancangan dan penerapannya. Berdasarkan latar belakang itulah, penelitian ini dilakukan agar dapat menghasilkan sebuah prototype perangkat lunak yang memberikan fungsi-fungsi dasar sebuah CASE tool untuk perancangan dan penerapan *object oriented database*, terutama dalam hal ini dikhususkan untuk Oracle 10g.

## 2. Alur Kerja

Ada 3 alur utama dalam aplikasi yang dibuat. Berikut ini adalah penjelasan lengkap ketiga alur tersebut :

### 2.1 Input

Pada tahap ini pengguna diminta untuk menggambar *class* pada diagram/kanvas yang telah disediakan aplikasi. Setelah menggambar *class* pengguna bias menambahkan atribut, method dan hubungan antar *class* seperti *Association*, *Generalization*, *Aggregation*, *Composition*. Kemudian aplikasi akan menyimpan semua informasi tentang *class*, atribut, method, serta hubungan antar *class* ke dalam sebuah variabel, yang berguna untuk alur kerja berikutnya.

---

<sup>(1)</sup> Titus Adi Prasetyo, Mahasiswa Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana

## 2.2 Proses

Proses utama yang dikerjakan oleh aplikasi ini adalah proses generate pemodelan yang dimodelkan oleh pengguna ke dalam bentuk SQL Oracle 10g. Proses ini diawali dengan membaca nilai variabel yang menyimpan semua informasi tentang *class*, atribut, method, serta hubungan antar *class*. Dari informasi yang diperoleh dari variabel tersebut maka proses generate dilakukan terhadap masing masing *class*. Adapun aturan utama dalam proses generate ada 2 yaitu :

- Jika terdapat hubungan *Generalisasi* maka *super class* harus di generate dahulu baru *sub class*.
- Jika terdapat hubungan bukan *Generalisasi* (*Association*, *Aggregation*, dan *Composition*) maka *sub class* harus di generate dahulu baru *super class*.

## 2.3 Output

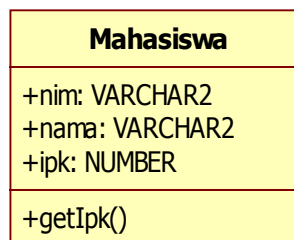
*Output* yang dihasilkan oleh aplikasi ini adalah berupa SQL hasil generate dari pemodelan yang dilakukan pengguna. Dari SQL yang dihasilkan pengguna dapat menyimpan SQL tersebut menjadi file .sql atau dapat langsung melakukan eksekusi terhadap SQL yang dihasilkan ke basis data Oracle 10g, yang terlebih dahulu dilakukan proses koneksi ke basis data Oracle 10g.

## 3. Model Konseptual *Object Relational Database*

Aplikasi alat bantu perancangan dan penerapan object oriented database ini mengikuti dasar-dasar dari sebuah model konseptual berbasis object.

### 3.1 Object Type, Atribut, dan Method

*Object type* merupakan bentuk implementasi pemodelan berbasis obyek di Oracle 10g yang relevan dengan UML *class diagram*. Sebagaimana sebuah *class* pada UML yang memiliki atribut dan method, *object type* juga memiliki atribut dan method. Berikut ini adalah contoh gambar sebuah *class* Mahasiswa yang diikuti dengan contoh SQL di Oracle 10g :



Gambar 1 Object Type Mahasiswa

```

//Deklarasi sebuah class
CREATE OR REPLACE TYPE Mahasiswa AS OBJECT
(
    //atribut
    Nim          VARCHAR2(8),
    Nama         VARCHAR2(30),
    Ipk          NUMBER,
    //method
    MEMBER FUNCTION getIpk RETURN NUMBER
);
/

//Type body yang mengimplementasikan method
CREATE TYPE BODY Mahasiswa AS
    MEMBER FUNCTION getIpk RETURN NUMBER IS
BEGIN
  
```

```

        RETURN Ipk;
    END;
END;
/

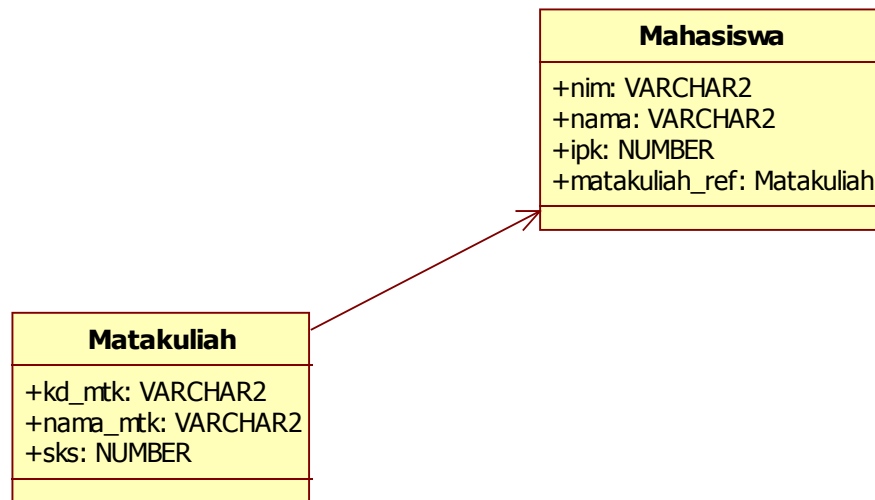
```

### 3.2 Hubungan Antar Class

Hubungan antar *class* yang ada dalam pemodelan berorientasi obyek di Oracle 10g antara lain *Association*, *Generalization*, *Aggregation*, serta *Composition*. Berikut ini penjelasan lengkap masing – masing hubungan diatas :

#### **Association**

Didalam pemodelan basis data berorientasi obyek di Oracle 10g, hubungan *Association* diimplementasikan menggunakan *reference object type*. Berikut ini contoh sebuah hubungan *Association* yang diikuti dengan betuk SQL di Oracle 10g :



Gambar 2 Association

```

CREATE OR REPLACE TYPE matakuliah AS OBJECT
(
    Kode          VARCHAR2(5),
    Nama_mtk      VARCHAR2(20),
    Sks           NUMBER
);
/

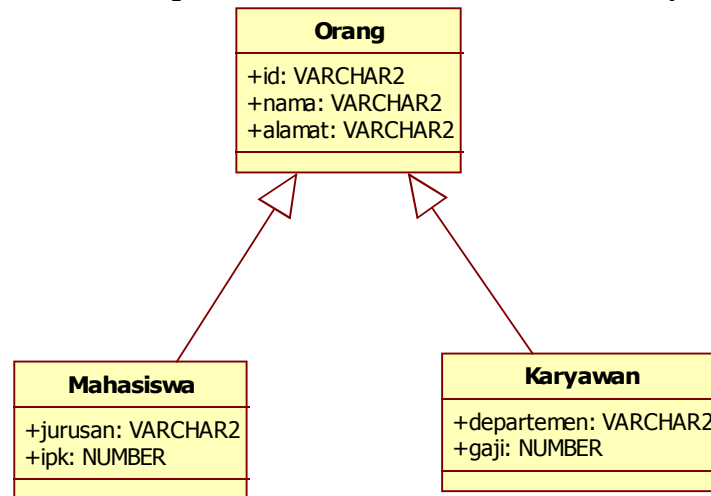
CREATE OR REPLACE TYPE mahasiswa AS OBJECT
(
    Nim           VARCHAR2(8),
    Nama          VARCHAR2(30),

    //Atribut reference sebagai penanda hubungan association
    Matakuliah_ref REF matakuliah
);
/

```

### Generalization

*Generalization* atau *Inheritance* atau pewarisan di dalam pemodelan basis data berorientasi obyek di Oracle 10g diimplementasikan menggunakan *keyword* NOT FINAL dan UNDER. Berikut ini contoh sebuah hubungan *Generalization* berserta contoh SQL nya di Oracle 10g :



Gambar 3 Generalization

```

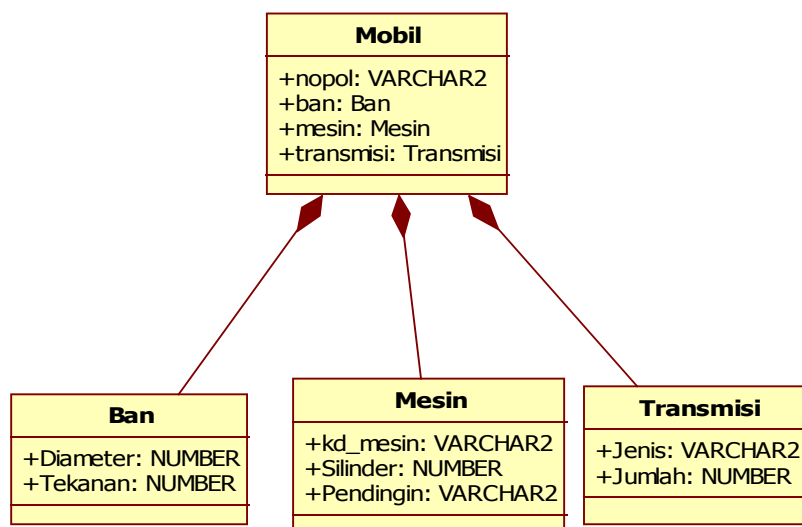
CREATE OR REPLACE TYPE Orang AS OBJECT
(
    id          VARCHAR2(10),
    nama        VARCHAR2(20),
    alamat      VARCHAR2(35)
)
//Not final sebagai penanda super class
NOT FINAL;
/

//Under sebagai penanda sub class
CREATE OR REPLACE TYPE Karyawan UNDER Orang
(
    departemen  VARCHAR2(10),
    gaji        NUMBER
);
/

//Under sebagai penanda sub class
CREATE OR REPLACE TYPE Mahasiswa UNDER Orang
(
    jurusan     VARCHAR2(10),
    ipk         NUMBER
);
/
  
```

### Composition

Didalam pemodelan basis data berorientasi obyek di Oracle 10g, hubungan *Composition* diimplementasikan menggunakan *composite object type*. Berikut ini contoh sebuah hubungan *Composition* yang diikuti dengan betuk SQL di Oracle 10g :



Gambar 4 Composition

```

CREATE OR REPLACE TYPE ban AS OBJECT
(
    Diameter      NUMBER,
    Tekanan       NUMBER
)
/

CREATE OR REPLACE TYPE mesin AS OBJECT
(
    Kd_mesin      VARCHAR2(20),
    Silinder       NUMBER,
    Pendingin     VARCHAR2(10)
)
/

CREATE OR REPLACE TYPE transmisi AS OBJECT
(
    Jenis         VARCHAR2(10),
    Jumlah        NUMBER
)
/

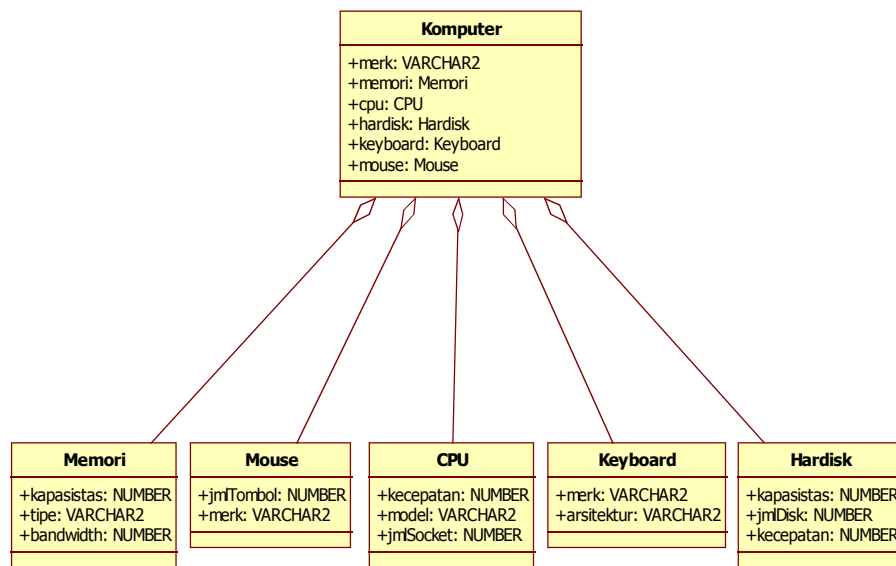
CREATE OR REPLACE TYPE mobil AS OBJECT
(

```

Nopol	VARCHAR2(7),
Ban	Ban,
Mesin	Mesin,
Transmisi	Transmisi
)	
/	

### Aggregation

Sama seperti hubungan *Composition*, hubungan *Aggregation* juga diimplementasikan menggunakan *composite object tipe*. Perbedaannya disini pada bentuk hubungan antar komponen dengan obyeknya. Jika pada hubungan *composition* hubungan antara komponen dan obyeknya bergantung penuh, maka pada hubungan *aggregation* hubungan antara komponen dan obyeknya adalah independent atau tidak bergantung. Berikut ini contoh hubungan *Aggregation* dilanjutkan dengan contoh bentuk SQL di Oracle 10g :



Gambar 5 Aggregation

```

CREATE OR REPLACE TYPE Memori AS OBJECT
(
    Kapasitas      NUMBER,
    Tipe           VARCHAR2,
    Bandwidth      NUMBER
)
/

CREATE OR REPLACE TYPE Mouse AS OBJECT
(
    jmlTombol      NUMBER,
    merk           VARCHAR2
)
/

CREATE OR REPLACE TYPE CPU AS OBJECT
(
    Kecepatan      NUMBER,
    jmlSocket      NUMBER,

```

```

        model          VARCHAR2
    )
/

CREATE OR REPLACE TYPE Keyboard AS OBJECT
(
    Merk          VARCHAR2,
    Arsitektur    VARCHAR2
)
/

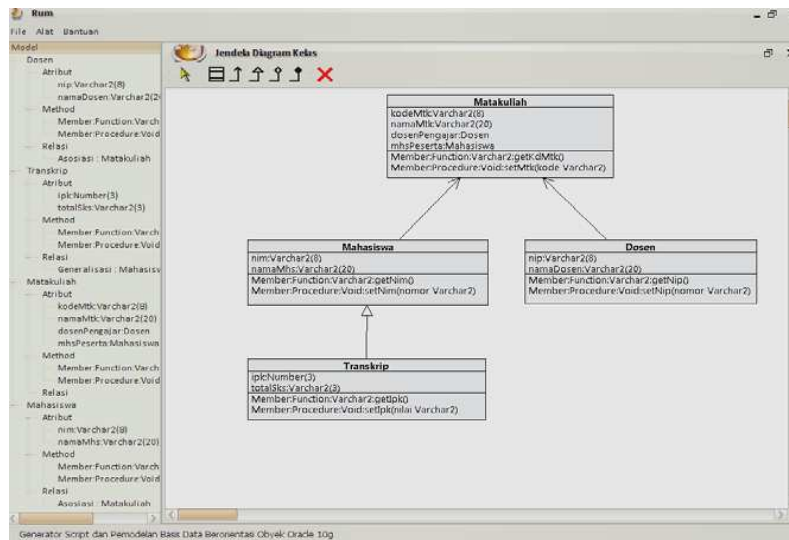
CREATE OR REPLACE TYPE Hardisk AS OBJECT
(
    Kapasitas          NUMBER,
    jmlDisk            NUMBER,
    kecepatan          NUMBER
)
/

CREATE OR REPLACE TYPE Komputer AS OBJECT
(
    Merk          VARCHAR2,
    CPU           CPU,
    Hardisk       Hardisk,
    Memori        Memori,
    Mouse         Mouse,
    Keyboard      Keyboard
)
/

```

#### 4. Pembahasan dan Analisis

Aplikasi alat bantu ini diberi nama dengan RUM. Pengembangannya menggunakan teknologi Java Desktop Swing. Gambar 6 menampilkan contoh tampilan dari aplikasi RUM ini. Untuk membantu pemakai dalam perancangan object relational database, RUM menyediakan fasilitas UML Class Diagram editor. Dengan editor ini, pemakai dapat menempatkan class-class yang mewakili entitas objek dari data yang akan disimpan, serta juga memodelkan hubungan antar class-nya.



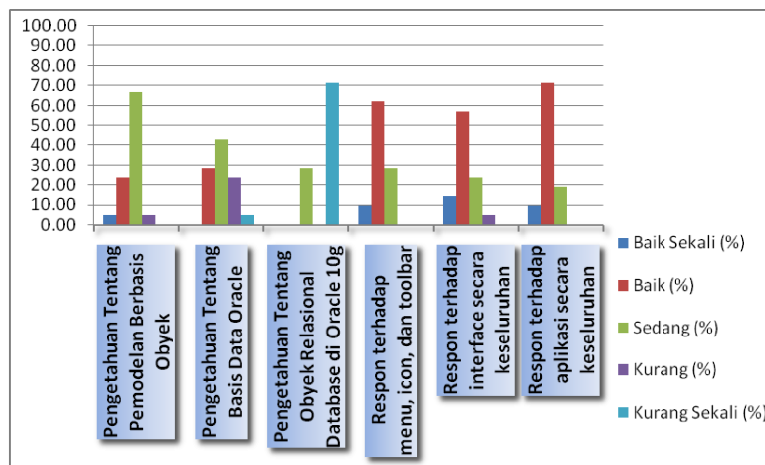
Gambar 6 Contoh Tampilan RUM

Dari class diagram UML yang dibuat, pemakai dapat melakukan eksport ke database Oracle. Dalam hal ini, aplikasi RUM akan melakukan konversi secara otomatis ke sintak SQL Oracle 10g berdasar skema class diagram yang sudah dibuat.

Untuk mengetahui seberapa mudah pemakaian aplikasi RUM ini, penulis melakukan analisis dengan menggunakan alat bantu kuesioner yang di berikan ke pada beberapa tester untuk mencoba menggunakan RUM. Dari kuesioner yang telah di berikan, berikut ini adalah tabel hasil kuesioner dan grafik dari tabel tersebut :

Tabel 1. Hasil Kuesioner

No	Kuesioner	Point					Jumlah (%)
		Baik Sekali (%)	Baik (%)	Sedang (%)	Kurang (%)	Kurang Sekali (%)	
1	Pengetahuan Tentang Pemodelan Berbasis Obyek	4.76	23.81	66.67	4.76	0.00	100.00
2	Pengetahuan Tentang Basis Data Oracle	0.00	28.57	42.86	23.81	4.76	100.00
3	Pengetahuan Tentang Obyek Relasional Database di Oracle 10g	0.00	0.00	28.57	0.00	71.43	100.00
4	Respon terhadap menu, icon, dan toolbar	9.52	61.90	28.57	0.00	0.00	100.00
5	Respon terhadap interface secara keseluruhan	14.29	57.14	23.81	4.76	0.00	100.00
6	Respon terhadap aplikasi secara keseluruhan	9.52	71.43	19.05	0.00	0.00	100.00



Gambar 7. Grafik Hasil Kuesioner

Dari data kuesioner yang diperoleh, latar belakang pengetahuan pengguna tentang pemodelan berbasis obyek adalah sebagian besar sedang yaitu 66,66 %, dan latar belakang pengetahuan tentang basis data Oracle sebagian besar sedang yaitu 42,85 %, tetapi disini latar belakang pengetahuan tentang obyek relasional database di Oracle 10g sangat kurang yaitu sebesar 71,42 % dari seluruh pengguna. Jadi bisa disimpulkan bahwa dari 100% pengguna yang melakukan testing Rum latar belakang pengetahuan tentang pemodelan berbasis obyek dan basis data Oracle 10g adalah dalam kategori sedang.



Dari sisi respon pengguna terhadap Rum, respon terhadap menu, icon, dan toolbar sebagian besar adalah baik yaitu 61,9 % dan untuk respon terhadap interface secara keseluruhan juga sebagian besar baik yaitu 57,14 % sedangkan untuk respon terhadap pengembangan aplikasi ini sebagian besar juga baik yaitu 71,42 %. Jadi bisa disimpulkan bahwa dari 100% pengguna yang melakukan testing Rum memiliki respon baik terhadap pengembangan Rum secara menyeluruh.

## 5. Kesimpulan

Dari hasil implementasi dan analisis sistem dalam penelitian ini maka dapat diambil beberapa kesimpulan sebagai berikut :

1. SQL yang dihasilkan dari proses generate oleh Rum tidak memiliki perbedaan dengan SQL yang dihasilkan secara manual, hanya saja terjadi sedikit perbedaan pada urutan SQL pembuatan kelas.
2. Pengguna Rum yang memiliki latar belakang pengetahuan tentang Pemodelan Berbasis Obyek dan Basis Data Oracle 10g, akan sangat membantu dalam menggunakan aplikasi Rum ini.
3. Dari sisi interface, Rum memiliki kekurangan terlihat dari hasil kuesioner yang menyatakan interface baik dan interface sedang nilainya cukup berimbang.

## 6. Daftar Pustaka

- Bales, Donald. (2002). *Java Programming with Oracle JDBC*. O'Reilly.
- Cole, Brian., Eckstein, Robert., Elliott, James., Loy, Marc., dan Wood, David. (2002). *Java Swing, 2nd Edition*. O'Reilly.
- Deitel, H, M.,(2004). *Java How to Program, Sixth Edition*. Prentice Hall.
- Greenwald, Rick., Stackowiak, Robert., Dodge, Gary., Klein, David., Shapiro, Ben., G. Chelliah, Christopher. (2005) *Profesional Oracle Programming*. Indianapolis. Wiley Publishing, Inc.
- Java Developer's Guide*. California. Oracle Corporation.
- JDBC Developer's Guide and Reference*. California. Oracle Corporation.
- M. Menon, R. (2005). *Expert Oracle JDBC Programming*. California. Appres.
- Object-Relational Features*. Oracle Database Application Developer's Guide .California. Oracle Corporation.
- Richardson, W. Clay., Avondolio, Donald., Schrager, Scot., W. Mitchell, Mark., dan Scanlon, Jeff. ( 2007 ). *Professional Java JDK 6 Edition*. Indianapolis. Wiley Publishing, Inc.
- Wenny Rahayu, Johanna., Taniar, David., dan Pardede, Eric. (2006) *Object-Oriented Oracle*. Hershey, London, Melbourne, Singapore. IIR Press